PROBLEM DETERMINATION USING PROBING

Field of the Invention

This invention generally relates to problem determination in a server or network element and, more specifically, to the deployment of probing technology for the purpose of problem determination.

Background of the Invention

A list of references is set forth at the close of the present disclosure; individual references from the list are referred to herebelow numerically (i.e., [1], [2], [3], etc...).

As networks continue to grow in size and complexity, system administrators are faced with an ever-increasing volume of event data, and tasks such as fault localization and problem determination become more difficult. As a result, tools are needed that can assist in performing these management tasks by responding quickly and accurately to the large number of events and alarms that are usually generated by even a single fault. Currently this task is usually done using event correlation techniques that utilize network dependency models to integrate the information provided by multiple alarms in order to determine the most likely cause of the problem. [3,4,5,6]

Probing technology is widely used to measure the quality of network performance, often motivated by the requirements of service-level-agreements. A probe is a program that executes on a particular machine (called a probing station) by sending a command or transaction to a server or network element and measuring the response (See Figure 1). Examples of probing technology include the T.J. Watson EPP technology [1] and the Keynote product [2]. Probing offers the opportunity to develop an approach to problem determination that is more active than traditional event correlation and other methods.

Several decisions are needed to use probes in practice. First, probing stations must be selected at one or more locations in the network. Second, the probes must be configured – it must be decided which network elements to target and which station each probe should originate from. Configuring a probe set in order to perform fault localization requires certain trade offs; the objective is to obtain a probe set which is both small, thereby minimizing network load and the costs of storing the probe results, yet also provides wide coverage, in order to locate problems anywhere in the network.

Summary of the Invention

In accordance with at least one presently preferred embodiment of the present invention, there is broadly contemplated a system and method of using probing technology for the purpose of problem determination. Probes to be sent may be actively

selected in order to be able to diagnose problems of particular interest, which allows for greater flexibility and efficiency. The extra load imposed on a network by the use of probes is small and may be minimized through the selection of the algorithms to be used.

In summary, one aspect of the invention provides an apparatus for diagnosing problems in multiple node networks using probing technology, the apparatus comprising: an arrangement for determining a candidate probe set; an arrangement for determining which problems can be diagnosed by the candidate probe set; and an arrangement for finding small probe sets which can diagnose the same problems as the candidate probe set.

Another aspect of the present invention provides a method for diagnosing problems in multiple node networks using probing technology, the method comprising the steps of: determining a candidate probe set; determining which problems can be diagnosed by the candidate probe set; and finding small probe sets which can diagnose the same problems as the candidate probe set.

Furthermore, an additional aspect of the invention provides a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for diagnosing problems in multiple node networks using probing technology, said method comprising the steps of: determining a candidate

probe set; determining which problems can be diagnosed by the candidate probe set; and finding small probe sets which can diagnose the same problems as the candidate probe set.

For a better understanding of the present invention, together with other and further features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying drawings, and the scope of the invention will be pointed out in the appended claims.

Brief Description of the Drawings

Figure 1 illustrates probing technology.

Figure 2 shows an example of a probe going through a network.

Figure 3 shows an example of two probes and the resulting dependency matrix.

Figure 4 shows the various problem diagnoses that can result from sending out the two probes.

Figure 5 shows an example where a third probe has been added to the previous two.

Figure 6 illustrates the system architecture.

Figure 7 describes the algorithm for computing the initial dependency matrix.

Figure 8 provides a worked example of this algorithm.

Figure 9 shows the algorithm for computing the diagnosable problems.

Figure 10 provides a worked example for this algorithm.

Figure 11 provides an Exhaustive Search algorithm for finding the minimal probe set.

Figure 12 presents a Quick Search algorithm for finding a small (not necessarily minimal) probe set.

Figure 13 presents a Greedy Search algorithm for finding a smaller (but also not necessarily minimal) probe set

Description of the Preferred Embodiments

The present invention employs a novel approach for fault localization and problem determination in networks. The approach utilizes an active probing technology, rather than traditional approaches which correlate alarm events received from multiple

sources when a failure occurs. The problem is precisely formulated, a system architecture for addressing the problem in a modular way is described, and algorithms for finding small probe sets that are able to diagnose problems occurring anywhere in the network are implemented.

The problem is formulated in terms of a dependency matrix methodology that captures the interactions between the paths which the probes traverse. The first step is to compute the initial dependency matrix using information about which probe-stations and probes are available to be used. The second step is to determine using the dependency matrix which problems can be diagnosed using the complete set of probes. The third step is to find smaller probe sets which are able to diagnose the same set of problems as the initial probe set. Small probe sets are desirable to minimize the additional network load and data storage costs imposed by the use of probes.

Three preferred algorithms are provided for computing small probe sets. An exhaustive search algorithm will find the smallest possible probe set able to diagnose the full set of problems, but this is computationally expensive. A linear-time algorithm is provided which very quickly finds a small (but not necessarily minimal size) probe set. A quadratic-time algorithm is provided which finds a smaller (but still not necessarily minimal size) probe set. A system architecture is provided which facilitates the decision

of which algorithm to use depending on the desired tradeoff between computational cost and probe set size.

Figure 1 illustrates how probing technology works. It is important to note that the network model is quite general. For example, layering can be accommodated: if a web-server depends on TCP/IP running which depends on the box being up, this can be modeled as a node for the box with a link to TCP/IP from the box and a further link from TCP/IP to the web-server. Thus nodes may represent applications and links dependencies between those applications. Similarly, a node may represent a sub-network of many nodes whose inter-connections are unknown. In this case probing will determine that the problem lies somewhere in that sub-network, at which point some form of local system management (perhaps including local probing) may be used to pinpoint the problem.

Finding the minimal set of probes needed for fault localization requires providing answers to the following questions: (1) Which probes are available as "candidates" for use in a network? (2) Which faults can be successfully identified by a given set of probes? (3) What is the smallest number of probes that can identify the same collection of faults as a given set? The present invention addresses these issues and integrates the solutions obtained into a system for performing fault localization.

Suppose the network has n nodes. Each probe is represented as a binary string of length n, where a 1 in position j denotes that the probe passes through node N_j. Each probe that is sent out either returns successfully or fails to do so. If a probe is successful, then every node and link along its path must be up. Conversely, if a node or link is down then any probe passing through that node or link fails to return.

Figure 2 shows a probe going from N_1 through N_2 to N_5. If this probe (denoted by P_{15}, indicating the starting and ending nodes) fails, that indicates a problem with either N_1, N_2, or N_5. If P_{15} succeeds, that indicates that N_1, N_2, and N_5 are all ok, although there may be a problem with one of the other nodes that the probe does not pass through.

A set of r probes defines a dependency matrix D, where D(i,j)=1 if probe P_i passes through node N_j and D(i,j)=0 otherwise. D is an r-by-n matrix, where r is the number of probes and n the number of nodes.

Figure 3 shows an example of 2 probes and the resulting dependency matrix.

If r probes are sent out, their results provide a "signal" – a binary string of length r, each digit denoting whether or not that probe returned successfully. For example, in Figure 3 if only N_2 is down then P_{15} fails but P_{16} succeeds. Similarly if only N_5 is down then P_{15} fails but P_{16} succeeds. Thus these two failures result in the same signal, because their columns in the dependency matrix are identical; i.e. a failure in N_2 cannot be

distinguished, by these 2 probes, from a failure in N_5. Any problem whose column in the dependency matrix is unique generates a unique signal and as a result can be unambiguously diagnosed.

Figure 4 shows the 4 possible signals that might result from sending out these 2 probes, and which problems might be the cause of each signal.

A failure in node N_1 can be uniquely diagnosed, because both probes fail and no other single node failure results in the same signal; N_1's column in the dependency matrix is unique. However, as explained above, a failure in N_2 cannot be distinguished from a failure in N_5, and similarly a failure in N_3 cannot be distinguished from a failure in N_6. Although N_4's column is unique, a failure in N_4 cannot be distinguished from no failure anywhere in the network, because there is no probe passing through N_4. Adding an extra "node" whose column is all zeroes, representing no failure, avoids this technicality. Any problem whose column in the dependency matrix is unique generates a unique signal and as a result can be unambiguously diagnosed.

Thus a dependency matrix decomposes the network into a disjoint collection of nodes, where each group consists of the nodes whose columns are identical; i.e., each group contains those nodes whose failure cannot be distinguished from one another by the given set of probes. This defines the diagnosable problems of a set of probes. For

example, in Figure 4 the diagnostic problems are {{1}, {2,5}, {3,6}, {4,NF}}, where index j represents node N_j and NF is the extra "node" representing no failure anywhere in the network. A failure is diagnosable if and only if its group is a singleton set.

Figure 5 shows an example where a third probe has been added to the previous 2. This third probe, P_{42}, goes from N_4 through N_3 to N_2. Note that these 3 probes are enough to diagnose a failure in any of the 6 nodes, because each column in the dependency matrix is unique.

Figure 6 illustrates the system architecture. First the candidate probes are identified, 610, and the dependency matrix and its diagnosable problems are determined, 620. Then a subset of the candidate probes is found which has the same diagnosable problems as the entire set. To do this first the diagnosable problems are found, 630. Various algorithms are available for computing the final probe set, depending on whether the user wants the guaranteed minimal set of probes or if a small but non-minimal set is adequate. The user selects the algorithm, 640, and the final set of probes is then output, 650. The dependency matrix can be stored in a database as an intermediate step, 660, so that later computations can make use of it without having to repeat earlier computations.

Determining the Initial Dependency Matrix

The architecture described above allows the set of candidate probes to be provided from whatever sources are available; for example a human expert may specify which probes are possible. It may also be useful, however, to compute the available probes from the network structure and the location of the probe stations. This provides a specific instantiation of step 610.

From the n nodes a subset of k nodes are selected as the probe stations. Potentially any node may be a probe station, but in practice only a small number are used; they are usually chosen based on extraneous considerations, such as controlled access to the machines. A probe can be sent to any node from any probe station. Thus the candidate set of probes could theoretically contain a probe for every possible route between every probe station and every node. In practice it cannot be guaranteed that a probe follows a particular path through the network, and thus routing strategies restrict the set of available probes; for example a probe may follow the shortest (i.e., least-cost) path through the network. This creates a candidate set of probes whose size r is a linear function of the number of nodes n; this set is sufficient to diagnose any single node being down because one can simply use one probe station and send a probe to every node.

Figure 7 describes the algorithm for computing the initial dependency matrix. For each probe-station S_i and each node N_j, a row of the dependency matrix corresponds to the shortest path through the network from S_i to N_j; each value in this row is set to either 1 or 0 depending on whether or not the path passes through that node.

Figure 8 gives an example of the output of this algorithm for the network shown there. There are 2 probe-stations, N_1 and N_4, and one probe from each probe-station to each node, following shortest path routing.

Determining the Diagnosable Problems of a Set of Probes

Given a dependency matrix, the decomposition into diagnosable problems places all problems with the same column into the same group. The decomposition is incrementally computed row-by-row. The key is that adding a row (i.e., a probe) always results in a more extensive decomposition, because nodes in distinct groups remain distinguishable; an additional probe can only have the effect of distinguishing previously indistinguishable nodes.

Figure 9 shows the algorithm for computing the diagnosable problems from a given dependency matrix. The algorithm starts with the decomposition {1,2, ..., n} - all nodes are indistinguishable, 910. Each additional probe decomposes every group of the current decomposition, 920, into two subgroups depending on which nodes the probe

passes through, 930, and all the nonempty subgroups are collected together, 940. This process is repeated for each probe, 950. Thus each of the nodes remains grouped with precisely those nodes it has not yet been distinguished from. The algorithm terminates with the complete decomposition after considering each probe only once.

Figure 10 illustrates how the diagnosable problem algorithm works for the set of 3 probes shown previously in Figure 5.

Finding the Minimal Set of Probes

The disclosure now turns to a discussion of finding the minimal set of probes that has the same diagnostic power as a given set. For example, the 3 probes shown in Figure 5 is the smallest subset of the full set of candidate probes shown in Figure 8 that is able to diagnose any single node failure. Clearly the minimal set of probes may not be unique, although the minimal number of probes is. Three algorithms for finding the minimal probe set are now examined: an exponential time exhaustive search and two approximation algorithms – one requiring linear time and the other quadratic time.

Figure 11 provides an Exhaustive Search algorithm for finding the minimal probe set. Since a node can only be diagnosed if there is at least one probe passing through it, probes can be added incrementally in all feasible combinations until the minimal set is reached. The computational complexity of this algorithm increases exponentially as the

number of candidate probes increases - this is clearly prohibitive unless the network is quite small.

Two approximation algorithms that heuristically attempt to find the minimum set, but are not guaranteed to do so, are now presented.

Figure 12 presents Quick Search - starting with the initial set of *r* probes, consider each probe in turn and discard it if it is not needed; i.e., a probe is discarded if the diagnosable problems remain the same even if it is dropped from the probe set. This process terminates in a subset with the same diagnostic power as the original set, but this subset may not necessarily be minimal. The complexity of this algorithm increases linearly in the size of the original probe set.

Figure 13 presents Greedy Search - starting with an empty set of probes, at each step add what looks like the "best" probe among the remaining probes. This process terminates in a subset with the same diagnostic power as the original set, but this subset may not necessarily be minimal. The complexity of this algorithm increases quadratically in the size of the original probe set. Greedy Search usually finds a smaller subset than Quick Search, but takes a little longer in computation time. For large networks the probe sets found by these algorithms are quite close to the true minimal probe set.

It is to be understood that the present invention, in accordance with at least one presently preferred embodiment, includes an arrangement for determining the candidate probe set, an arrangement for determining which problems can be diagnosed by the candidate probe set, and an arrangement for finding small probe sets which can diagnose the same problems as the candidate probe set, all of which may be implemented on at least one general-purpose computer running suitable software programs. These may also be implemented on at least one Integrated Circuit or part of at least one Integrated Circuit. Thus, it is to be understood that the invention may be implemented in hardware, software, or a combination of both.

If not otherwise stated herein, it is to be assumed that all patents, patent applications, patent publications and other publications (including web-based publications) mentioned and cited herein are hereby fully incorporated by reference herein as if set forth in their entirety herein.

Although illustrative embodiments of the present invention have been described herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention.

LIST OF REFERENCES

[1] A. Frenkiel and H. Lee. EPP: A Framework for Measuring the End-to-End Performance of Distributed Applications. Proceedings of Performance Engineering 'Best Practices' Conference, IBM Academy of Technology, 1999.

[2] "Using Keynote Measurements to Evaluate Content Delivery Networks", available at [http://]www.keynote.com/services/html/product_lib.html

[3] I. Katzela and M. Schwartz. Fault Identification Schemes in Communication Networks, IEEE/ACM Transactions on Networking, 1995.

[4] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, S. Stolfo. A Coding Approach to Event Correlation, IM 1997.

[5] B. Gruschke. Integrated Event Management: Event Correlation Using Dependency Graphs, DSOM 1998.

[6] C.S. Hood and C. Ji. Proactive network fault detection. Proceedings of INFOCOM, 1997.

[7] Patent: US05528516 - Apparatus and method for event correlation and problem reporting. Date Issued: 06/18/1996

[8] Patent: US05661688 - Apparatus and method for analyzing and correlating events in a system using a causality matrix. Date Issued: 08/26/1997